Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

Using protein-domain information for multiple sequence alignment

Layal Al Ait, Eduardo Corel, Burkhard Morgenstern
University of Göttingen, Institute of Microbiology and Genetics, Department of Bioinformatics
Goldschmidtstr. 1, 37077 Göttingen, Germany
{layal,eduardo,burkhard}@gobics.de

***Abstract*—Most approaches to multiple sequence alignment rely on primary-sequence information. External sources of information, however, can give valuable hints to possible sequence homologies that may not be obvious from sequence comparison alone. Given the huge amount of sequence annotation that is being produced on a daily basis, integrating such external information into the alignment process can contribute to produce biologically more meaningful alignments. In this paper, we investigate different approaches to use existing information about protein domains for improved multiple alignments. We use the *PFAM* database to identify possible domains in protein sequences, and we use this information to align protein sequences with *DIALIGN* and with a recently developed graph-theoretical approach to multiple alignment. Test runs on *BAliBASE* and *SABmark* show that this approach leads to improved alignments.**

***Index Terms*—Multiple sequence alignment, protein domains, anchored alignment.**

I. Introduction

Multiple sequence alignment (MSA) is a pivotal tool in biological sequence analysis. With the ever increasing amount of sequence data, algorithm development for MSA remains a central field of research in bioinformatics, see for example [1, 2] for recent reviews. Most algorithms for protein alignment are based on primary sequence information alone. On the other hand, a large amount of additional information and biological data is being provided daily for public use, and numerous software tools and databases are available to annotate protein sequences. Whether this data is about newly discovered protein domains, or recently identified protein features and functions, one should take advantage of this available data and try to integrate it in algorithms for MSA. One of the rich public resources about protein sequences is Interpro [3]. It has eleven member databases, one of which is PFAM [4], a large collection of multiple sequence alignments and Hidden Markov Models (HMMs) covering many common protein domains and families. Adding such a source of information should improve the multiple alignments.

A number of approaches to MSA have been developed following these ideas. *Cobalt* [5], for example, uses external constraints derived from data base searches, sequence similarity and user input. *T-Coffee* [6] uses *primary libraries* of pairwise alignments as a starting point for multiple alignment. By default, these alignments are calculated by standard global and local alignment algorithms. It is possible, however, to include arbitrary user-defined alignments into the primary libraries, to use other sources of information. This has been used, for example, to integrate structural information for multiple protein alignment [7]. *DbClustal* [8] is a web application that allows to include external information retrieved from database searches. *DIALIGN* [9] has an *anchoring option* [10, 11] that allows users to enforce the alignment of specific regions of the sequences. This option can be used if previous knowledge about homologous regions of the sequences is available. More recently, the developers of *Clustal Omega* [12] included an option to their program that allows the user to supply a *profile Hidden Markov Model* [13] to guide the multiple alignment procedure.

In this paper, we describe an approach to protein MSA that uses hits to *PFAM* domains as hints for possible homologies. Our approach is inspired by the *external profile alignment (EPA)* approach implemented in *Clustal Omega*. The idea is, to search the input sequences against a database of known protein domains; sequence positions matching the same positions of these domains are considered as potential homologues. We assemble *blocks* of equal-length segments of the input sequences matching to the same position of a *PFAM* domain. Possible local homologies obtained in this way are combined with local sequence similarities found by the program *DIALIGN* to produce a final multiple alignment.

II. Algorithm

A. Scanning sequences against PFAM

Each protein family in *PFAM* is represented by a *model* consisting of one or several multiple sequence alignments of *domains* and Hidden Markov Models (HMM) derived from these alignments. Thus, the first step in our approach is to detect common domains in a set of sequences and then aligning these domains together. In order to scan the input sequences against *PFAM* we use *HMMER* [13, 14]. More precisely, we use the program *Hmmscan* which searches sequences against a given profile HMM database.

HMMER assigns quality scores to matches between sequences and models of proteins and domains in a database. In order to control which hits are used by our algorithm, we use two threshold values for E-values of *HMMER* hits. The first threshold E_m concerns the *E-value* of the matched models and ensures that only models with *E-value* less than E_m are taken into consideration. Those profiles which satisfy the first

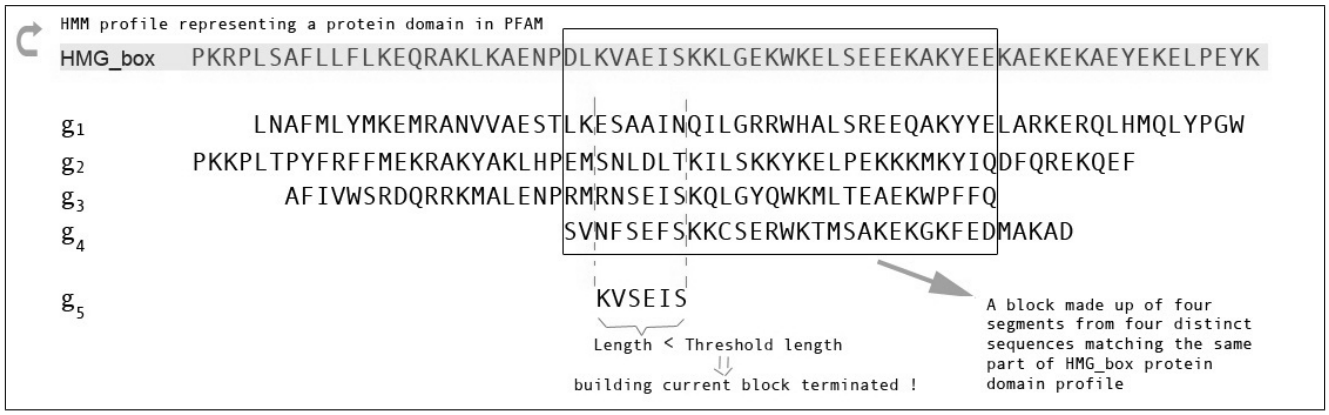

Fig. 1. The sequence in the grey box represents a part of the HMM profile for the HMG_box protein domain. A set of input sequences were scanned against PFAM. Four Segments, $g_1 \ldots g_4$, belonging to distinct sequences were found to match the same protein profile. The overlapping parts between the four segments and the HMM profile produce a block of length 28. If one would add a fifth segment g_5 which also matches the HMG_box profile, the length of the block would drops to 6, which is less than the length threshold $L_0 = 7$ that we are using. Thus, our algorithm does not add g_5 to the growing domain block.

threshold condition are further filtered with a second threshold E_d for domains. Only those domains that have an *E-value* less than E_d are considered in our procedure. In this study, we used the values of 5×10^{-3} for E_m and 10^{-4} for E_d. (Note that a model in Pfam can comprise more than one protein domain; our first threshold applies to full models, the second one to single domains).

For the *blocks* that we want to assemble from our *PFAM* matches, we need gap-free local alignments of our input sequences to *PFAM* domains. In general, however, matches of protein sequences to *PFAM* domains may contain gaps. In order to obtain gap-free matches, we extract the gap-free parts out of the obtained matches, provided they have a length of at least L_0, where L_0 is a user-defined threshold. For the results reported here, we used a length threshold of $L_0 = 7$, since after testing various values for L_0, the value of 7 was found out to give the best results.

For a segment g of an input sequence s that matches without gaps to a *PFAM* domain, we denote by $L(g)$ the length of g and by $\omega(g)$ the score of s to the PFAM domain as provided by HMMER. By G we denote the set of all segments g from our input sequences matching without gaps to a *PFAM* domain.

B. *Building* domain blocks *of* PFAM *matches*

The basis of our alignment approach are *blocks* of segments $g \in G$ of the input sequences matching to the same corresponding position of some *PFAM* domain. We call such a block a *domain block*. More precisely, a *domain block* consists of segments that are matched without gaps to the same segment of a *PFAM* domain. For each column in a domain block, the corresponding positions in the involved segments are required to match to the same position in the *PFAM* domain that is associated with this block.

To construct a domain block B, we start by including the segment $g \in G$ that has the highest score $\omega(g)$ into B. We continue by processing all segments in $g' \in G$ matching the same domain as g at the corresponding positions in decreasing order of their match scores $\omega(g')$. If the overlap of a new segment g' with B has at least a length of L_0, then g' is added to B and removed from G. Where necessary, B and/or the new segment g' are truncated to ensure that all segments in B have the same length (Fig.1).

Let $L(B)$ denote the length of the domain block B, *i.e.* the length of the segments contained in B. Since we ensure that all segments in B have the same length, the length of B may be reduced each time a new segment is added to B. To avoid that $L(B)$ shrinks below our length threshold L_0, we ignore all segments that would cause the length of B to drop below L_0 if they would be added to B.

It is possible that adding a segment to a block B would cause a relatively large sub-block B' of B to be chopped out, which means that some information is being lost. In this case – and if the length of the sub-block B' is above our threshold L_0 –, the segments of B' are added again to our set G and considered to assemble the next blocks. B' will either be prefix of B or a suffix, and we can have both cases occurring simultaneously.

After no more segments can be found that match to the current domain, the segment with the largest score ω remaining in the set G is taken to start the assembly of the next block as described above.

C. *Integrating domain blocks and primary-sequence similarity*

To integrate the *domain blocks* derived from *PFAM* hits with similarities at the primary-sequence level, we use the MSA program *DIALIGN*. Here, we apply two different approaches to combine our blocks with local alignments produced by *DIALIGN*.

1) First, we use the domain blocks as *anchor points* for *DIALIGN*. DIALIGN has an *anchoring* option, where users can specify local alignments as anchor points that should be preferentially aligned. Since, in general, not all selected anchor points can be included in one single output alignment, the program greedily selects a

consistent subset of the proposed anchor points, *i.e.* a subset fitting into one single multiple alignment, see [11] for details. By definition, *anchor points* in DIALIGN are pairs of residues that are to be aligned (or, more generally, pairwise alignments that are to be included into a multiple alignment). We therefore use pairs of segments contained in our *domain blocks* as anchor points for DIALIGN (to ensure that all segments of a domain block are connected directly or indirectly by anchor points, we define anchor points connecting segment 1 with segment 2, segment 2 with segment 3, segment 3 with segment 4 etc).

The scores of the anchor points derived from a block B is defined as the *sum* of the scores $\omega(g)$ of the segments g that are part of B. (The scores of anchor points determine their priority in the greedy selection of a *consistent* set of anchor points.) As a result, in this approach, we first align the segments that are part of the constructed domain blocks are aligned. The rest of the sequences is then aligned by *DIALIGN* under the constraints defined by the selected anchor points.

2) In an alternative approach, we consider all local similarities that are either represented in one of our *domain blocks* or are aligned in the respective optimal pairwise alignment calculated by *DIALIGN* [15, 16]. To integrate these similarities into one MSA, we use a previously developed graph-theoretical method described in [17]. In short, we construct a so-called *incidence graph* from our local similarities (Fig. 2). Here, positions of the input sequences are represented as nodes and there is an edge between two nodes if these two positions are aligned, either in a *domain block* or in one of the pairwise *DIALIGN* alignments.

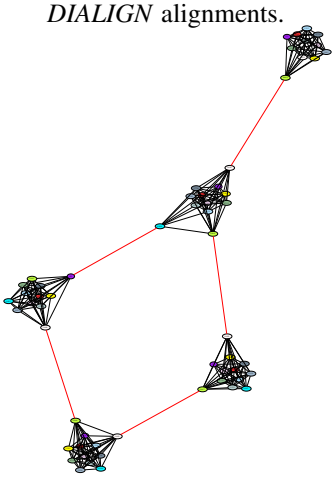

Fig. 2. Incidence graph of fragments: vertices are positions in sequences (color-coded) related by an edge if aligned by a fragment. When red edges are removed, connected components are partial alignment columns.

Typically, a set of homologous positions in a set of sequences would appear as a highly connected sub-graph of our incidence graph. Ideally, such a set of positions would also be a *connected component* of the incidence graph. Because of spurious similarities, however, there may also be edges connecting the subgraph with non-homologous positions of the sequences.

For a set of input sequences, we call a group P of sequence positions a *partial alignment column* if from each of the sequences at most one position is contained in P. (In this case, P could form a (partial) column in a multiple alignment). Our approach works by extracting highly connected partial alignment columns in our incidence graph. Such *partial alignment columns* are obtained by removing some edges of the graph. In [17], we proposed a *min-cut/max-flow* algorithm to extract highly connected partial alignment columns form the incident graph. Note that the partial columns that are constructed in this way need not be globally consistent, though. We applied therefore a previously developed algorithm that removes from these partial columns positions that are responsible for inconsistencies. This algorithm is described in detail in [18].

III. Testing and Results

To evaluate our approach, we ran four different versions of *DIALIGN*:

1) The standard version of the program, *DIALIGN 2.2.1*,
2) Our graph-theoretical approach using local pairwise alignments from the default version of *DIALIGN* [17].
3) *DIALIGN* with anchor points defined by *domain blocks*
4) Our graph-theoretical approach [17] using both *domain blocks* and local alignments from *DIALIGN*

We compared these four approaches with six established MSA programs: *ClustalW* 2.1 [19], *MAFFT* 6.903beta [20], *ProbCons* 1.12 [21], *MUSCLE* 3.8.31 [22], *T-Coffee* 5.31 and *COBALT 2.0.1*. Here, we ran *ClustalW, MUSCLE, COBALT, T-Coffee* and *ProbCons* with default parameters, while we used *MAFFT* with the *'linsi'* option. Testing our approach was done on two benchmark databases: *BAliBASE* [23, 24] and *SABmark* [25].

BAliBASE is a database of manually refined multiple sequence alignments specifically designed as reference alignments for the evaluation and comparison of multiple sequence alignment programs. It is composed of six datasets RV11, RV12, RV20, RV30, RV40 and RV50, categorized by sequence length, similarity, and presence of N/C terminal extensions. RV11 contains 38 equidistant families with sequence identity $< 20\%$, RV12 contains 44 equidistant families with sequence similarity between 20% and 40%. RV20 contains 41 families with similarity more than 40%. RV30 is comprised of sequences from protein families with some highly diverged sequences. RV40 contains 49 families with large N/C terminal extensions and finally RV50 contains sequences with large internal insertions with a total of 16 alignments. Each reference alignment in BAliBASE contains a number of *core blocks* that are considered to be reliably aligned. We used the application bali_score provided by *BAliBASE 3.0* in order to calculate the scores of the alignments produced by our approach. There are two scores that evaluate a *test alignment* with a *reference*

TABLE I

PERFORMANCE OF DIFFERENT ALIGNMENT PROGRAMS ON THE *BAliBASE* BENCHMARK DATABASE. *DIALIGN-cpc* USES OUR GRAPH-THEORETICAL APPROACH [17] BASED ON SEQUENCE SIMILARITY ALONE. *DIA+PFAM* IS *DIALIGN* USING *PFAM* HITS AS ANCHOR POINTS, WHILE *DIA-PFAM-mix-cpc* IS OUR GRAPH-THEORETICAL APPROACH USING SIMILARITIES FOUND BY *DIALIGN* TOGETHER WITH *PFAM* HITS.

Aligner	RV11		RV12		RV20		RV30		RV40		RV50	
	SP	TC	SP	TC	SP	TC	SP	TC	SP	TC	SP	TC
ClustalW	50.06	22.99	86.43	80.89	85.16	22.16	72.49	27.59	78.93	39.82	74.25	31.15
T-Coffee	57.97	31.61	92.49	82.1	90.97	38.23	79.65	36.79	89.19	54.81	89.39	58.88
COBALT	46.66	**61.47**	79.28	**90.63**	75.0	**88.73**	66.97	**79.96**	83.21	43.66	82.41	45.94
ProbCons	**66.97**	41.96	**94.12**	86.04	91.6	41.14	84.52	54.72	90.33	53.61	89.41	57.88
Muscle	57.15	32.06	91.53	80.89	88.91	35.30	81.44	41.19	86.48	45.31	83.51	46.38
MAFFT	65.31	43.14	93.55	84.31	**92.53**	45.12	**85.9**	58.52	**91.54**	**59.43**	**90.09**	**59.85**
Dialign	50.7	26.8	86.6	70.0	86.9	29.7	74.0	31.6	83.3	44.5	80.7	42.9
Dialign-cpc	53.1	30.5	88.5	74.0	90.5	35.2	78.6	41.0	87.7	50.2	86.5	50.9
Dia+pfam	58.6	38.4	86.3	71.6	88.8	36.9	78.5	42.2	86.1	48.2	80.1	45.9
Dia+pfam-mix-cpc	54.7	32.6	89.7	77.3	90.3	34.2	75.1	37.8	87.5	50.1	86.3	51.3

alingment of the same sequences, the sum-of-pairs (SP) score and the true-columns score (TC).

The *SP score* is the percentage of residue pairs in the *core blocks* of the reference alignment that are also correctly aligned in the test alignment. Similarly, the the *TC score* is the percentage of columns in the core blocks of the reference alignment that are also correctly aligned in the test alignment.

SABmark is an automatically generated benchmark database for multiple protein alignment containing sequences from the *SCOP* data base [26]. It is composed of two large sets of benchmark alignments, the *twilight zone* and the *superfamilies* set. The twilight zone set contains 209 groups of sequences that share $< 25\%$ identity. The superfamilies set consists of 425 groups of sequences with common evolutionary origin, they share almost 50% identity. For testing against *SABmark*, we used the *fp* score as a criterion. This score is equivalent to the *sp score* used on *BAliBASE*. In addition, we used the so-called and the *modeler score 'fm'* [27], defined as the number of correctly aligned residue pairs found in the test alignment divided by the total number of aligned residue pairs in the test alignment. Table I shows the results of the compared alignment methods on *BAliBASE*. Table II shows the testing results on *SABMark*.

Runnig Time: Considering set RV11 in BAliBASE which consists of 38 sequences files, *DIALIGN* took 32 seconds to align the whole set. On the other hand, using our approach, it took 7.5 minutes to align the whole set. This time is distributed as follows: 66 seconds for the extraction of protein domains from PFAM, 7.3 minutes for building up the blocks and producing anchor points and 18.6 seconds to align the sequences with *DIALIGN* using the already produced anchor points. It can be noted that the running time of *DIALIGN* dropped from 32 seconds to 18.6 seconds due to the use of the anchor points.

IV. CONCLUSION

Most methods for multiple protein alignment are based on primary-sequence similarity alone. In this study, we investigated how matches of protein sequences to known protein *domains* can identify homologous residues in these sequences that can be used as an additional source of information in the construction of a multiple alignment. To obtain groups of possibly homologous positions, we matched the input sequences against *PFAM* and identified segments from the input sequences matching to the same positions in some *PFAM* domain.

TABLE II

PERFORMANCE OF THE ALIGNMENT PROGRAMS ON *SABmark*. NOTATION AS IN TABLE I.

Aligner	twi		sup	
	fd	fm	fd	fm
ClustalW	22.46	14.99	50.69	38.03
T-Coffee	23.61	17.9	52.53	41.26
Cobalt	**30.31**	20.64	**58.63**	44.18
ProbCons	28.878	20.8649	56.632	43.5215
Muscle	23.98	16.49	52.756	39.8179
MAFFT	26.432	19.06	55.37	42.487
Dialign	18.707	18.557	46.01	42.354
Dialign-cpc	18.860	19.885	46.486	44.440
Dia+pfam	22.216	**22.808**	48.86	**46.748**
Dia+pfam-mix-cpc	20.452	22.369	47.548	46.696

To date, only few MSA programs can include external information in addition to primary sequences. We used our program *DIALIGN* since this program has an *anchoring* option allowing the user to specify positions of the input sequences that are to be aligned. In a first approach, we directly used pairs of segments matching to the same positions in *PFAM* as *anchor points* for *DIALIGN*. Thus, these matches to *PFAM* were given priority in the alignment process, the remainder of the sequences was then aligned by *DIALIGN* respecting the constraints given by the *PFAM* matches. In a second approach, we combined matches to *PFAM* with primary-sequence similarities obtained by the standard version of *DIALIGN*. To this end, we used a recently published *graph-theoretical* approach [17].

On the benchmark database *BAliBASE*, *DIALIGN* with anchor points defined by *PFAM* matches performed consistently better than the standard version of the program that uses primary-information alone. However, there was no clear winner among *DIALIGN* with domain-based anchors, the graph-

theoretical method using primary sequence alone and the graph method based combining primary-sequence information *and* hits to *PFAM* domains. The best performing programs on *BAliBASE* were still *ProbCons, COBALT* and *MAFFT*. This is not surprising, as *BAliBASE* primarily contains *globally* related sequences, *i.e.* protein sequences related over their entire lengths. *ProbCons* and *MAFFT* are focusing on global alignment, while *DIALIGN* is based on local sequence similarities.

On *SABmark*, the winners were *COBALT* and *DIALIGN* using hits to *PFAM* domains. *COBALT* had the highest *sensitivity* scores while *DIALIGN* had the highest scores for *specificity*. *DIALIGN*, considers local similarities – or *PFAM* hits in our domain approach – for alignment that show some statistically significant similarity.

In our study, we used the *anchoring* option of *DIALIGN* and our graph-theoretical approach – also in combination with anchored *DIALIGN* – since these two alignment approaches can easily integrate arbitrary external information about possible homologies among the input sequences. In principle, it should be possible to adapt other MSA methods in a similar way, *e.g.* by adding a term for external homology information to the commonly used substitution scores. This should be particularly interesting for *probabilistic* methods such as *Probcons*, *MUMMALS [28], CONTRAlign [29]*, or *Probalign* [30]. Here, it would be possible, to combine primary-sequence information with external information in a single probabilistic model, similar as it has been done in the program *AUGUSTUS* for gene prediction in eukaryotes [31, 32].

Availability and Requirements

The source code of our method is available on request.

Acknowledgment

The authors would like to thank Dr. Thomas Lingner for helpful discussions.

References

[1] R. C. Edgar and S. Batzoglou, "Multiple sequence alignment," *Current Opinion in Structural Biology*, vol. 16, pp. 368–373, 2006.

[2] C. Notredame, "Recent evolutions of multiple sequence alignment algorithms," *PLOS Computational Biology*, vol. 3, p. e123, 2007.

[3] S. Hunter, P. Jones, A. Mitchell, R. Apweiler, T. K. Attwood, A. Bateman, T. Bernard, D. Binns, P. Bork, S. Burge, E. de Castro, P. Coggill, M. Corbett, U. Das, L. Daugherty, L. Duquenne, R. D. Finn, M. Fraser, J. Gough, D. Haft, N. Hulo, D. Kahn, E. Kelly, I. Letunic, D. Lonsdale, R. Lopez, M. Madera, J. Maslen, C. McAnulla, J. McDowall, C. McMenamin, H. Mi, P. Mutowo-Muellenet, N. Mulder, D. Natale, C. Orengo, S. Pesseat, M. Punta, A. F. Quinn, C. Rivoire, A. Sangrador-Vegas, J. D. Selengut, C. J. A. Sigrist, M. Scheremetjew, J. Tate, M. Thimmajanarthanan, P. D. Thomas, C. H. Wu, C. Yeats, and S.-Y. Yong, "Interpro in 2011: new developments in the family and domain prediction database," *Nucleic Acids Research*, vol. 40, pp. D306–D312, 2012.

[4] R. Finn, J. Tate, J. Mistry, P. Coggill, J. Sammut, H. Hotz, G. Ceric, K. Forslund, S. Eddy, E. Sonnhammer, and A. Bateman, "The pfam protein families database," *Nucleic Acids Res.*, vol. 36, pp. D281–D288, 2008.

[5] J. S. Papadopoulos and R. Agarwala*, "Cobalt: constraint-based alignment tool for multiple protein sequences," *Bioinformatics*, vol. 23, pp. 1073–1079, 2007.

[6] C. Notredame, D. Higgins, and J. Heringa, "T-Coffee: a novel algorithm for multiple sequence alignment," *J. Mol. Biol.*, vol. 302, pp. 205–217, 2000.

[7] O. Poirot, K. Suhre, C. Abergel, E. O'Toole, and C. Notredame, "3DCoffee@igs: a web server for combining sequences and structure into a multiple sequence alignment," *Nuc. Acids. Res*, vol. 32, pp. W37–W40, 2005.

[8] J. D. Thompson, F. Plewniak, J.-C. Thierry, and O. Poch, "DbClustal: rapid and reliable global multiple alignments of protein sequences detected by database searches," *Nucleic Acids Research*, vol. 28, pp. 2919–2926, 2000.

[9] B. Morgenstern, A. Dress, and T. Werner, "Multiple DNA and protein sequence alignment based on segment-to-segment comparison," *Proc. Natl. Acad. Sci. USA*, vol. 93, pp. 12 098–12 103, 1996.

[10] B. Morgenstern, N. Werner, S. J. Prohaska, R. S. I. Schneider, A. R. Subramanian, P. F. Stadler, and J. Weyer-Menkhoff, "Multiple sequence alignment with user-defined constraints at GOBICS," *Bioinformatics*, vol. 21, pp. 1271 – 1273, 2005.

[11] B. Morgenstern, S. J. Prohaska, D. Pöhler, and P. F. Stadler, "Multiple sequence alignment with user-defined anchor points," *Algorithms for Molecular Biology*, vol. 1, p. 6, 2006.

[12] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Sding, J. D. Thompson, and D. G. Higgins, "Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega," *Molecular Systems Biology*, vol. 7, p. 539, 2011.

[13] S. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, pp. 755–763, 1998.

[14] R. Finn, J. Clements, and S. Eddy, "HMMER web server: interactive sequence similarity searching," *Nucleic Acids Res.*, vol. 39, pp. W29–W37, 2011.

[15] B. Morgenstern, "A space-efficient algorithm for aligning large genomic sequences," *Bioinformatics*, vol. 16, pp. 948–949, 2000.

[16] ——, "A simple and space-efficient fragment-chaining algorithm for alignment of DNA and protein sequences," *Applied Mathematics Letters*, vol. 15, pp. 11–16, 2002.

[17] E. Corel, F. Pitschi, and B. Morgenstern, "A min-cut algorithm for the consistency problem in multiple sequence alignment," *Bioinformatics*, vol. 26, pp. 1015–1021, 2010.

[18] F. Pitschi, C. Devauchelle, and E. Corel, "Automatic detection of anchor points for the multiple alignment of biological sequences," *BMC Bioinformatics*, accepted for publication.

[19] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, pp. 4673–4680, 1994.

[20] K. Katoh, K. Misawa, K. Kuma, and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform," *Nuc. Acids Research*, vol. 30, pp. 3059 – 3066, 2002.

[21] C. B. Do, M. S. Mahabhashyam, M. Brudno, and S. Batzoglou, "ProbCons: Probabilistic consistency-based multiple sequence alignment," *Genome Research*, vol. 15, pp. 330–340, 2005.

[22] R. Edgar, "MUSCLE: Multiple sequence alignment with high score accuracy and high throughput," *Nuc. Acids. Res.*, vol. 32, pp. 1792–1797, 2004.

[23] J. D. Thompson, F. Plewniak, and O. Poch, "BAliBASE: A benchmark alignment database for the evaluation of multiple sequence alignment programs," *Bioinformatics*, vol. 15, pp. 87–88, 1999.

[24] J. D. Thompson, P. Koehl, R. Ripp, and O. Poch, "BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark," *Proteins: Structure, Function, and Bioinformatics*, vol. 61, pp. 127–136, 2005.

[25] I. V. Walle, I. Lasters, and L. Wyns, "SABmark - a benchmark for sequence alignment that covers the entire known fold space," *Bioinformatics*, vol. 21, pp. 1267 – 1268, 2005.

[26] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "Scop: a structural classification of proteins database for the investigation of sequences and structures." *J. Mol. Biol.*, vol. 247, no. 4, pp. 536–40, 1995.

[27] J. Sauder, J. Arthur, and J. R.L. Dunbrack, "Large-scale comparison of protein sequence alignment algorithms with structure alignments." *Proteins*, vol. 40, pp. 6–22, 2000.

[28] J. Pei and N. V. Grishin, "MUMMALS: multiple sequence alignment improved by using hidden markov models with local structural information," *Nucleic Acids Resarch*, vol. 34, pp. 4364–4374, 2006.

[29] C. B. Do, S. Gross, and S. Batzoglou, "CONTRAlign: Discriminative training for protein sequence alignment," in *Proceedings RECOMB'06*, 2006.

[30] U. Roshan and D. R. Livesay, "Probalign: multiple sequence alignment using partition function posterior probabilities," *Bioinformatics*, vol. 22, pp. 2715–2721, 2006.

[31] M. Stanke, O. Schöffmann, B. Morgenstern, and S. Waack, "Gene prediction in eukaryotes with a Generalized Hidden Markov Model that uses hints from external sources," *BMC Bioinformatics*, vol. 7, p. 62, 2006.

[32] M. Stanke, A. Tzvetkova, and B. Morgenstern, "AUGUSTUS+ at EGASP: using EST, protein and genomic alignments for improved gene prediction in the human genome," *Genome Biology*, vol. 7, p. S11, 2006.